



HP-UX System Administration 认证辅导丛书

UNIX

系统基础

版本 G.02

(美) HEWLETT-PACKARD COMPANY 著
陈 谦 等译



取自惠普专业培训教材
权威性的 UNIX 入门书籍
获得惠普技术认证的指定教材



机械工业出版社
China Machine Press

TP316.81

H87

HP-UX System Administration 认证辅导丛书

UNIX 系统基础 (版本 G.02)

(美) HEWLETT-PACKARD COMPANY 著

陈 谦 杨海俊 夏 阳 等 译



机械工业出版社

本书是 UNIX 的入门书籍。它以 HP-UX 为实例,系统介绍了 UNIX 的特点、基本命令、CDE、文件操作与权限、进程管理、管道、重定向等大量 SHELL 功能的使用和 SHELL 编程。实例丰富,极易学习和查阅。

本书内容丰富,条理清晰,适合系统管理人员,网络管理人员,以及广大 UNIX 初学者使用。本套丛书为惠普 IT 认证技术专家考试指定辅导书,可为广大希望取得惠普认证的人员提供参考。

本书为 HEWLETT-PACKARD COMPANY 公司所出英文版的授权译本。

版权所有。未经出版者批准,不得以任何形式或任何手段,电子的或机械的,包括照相、复制、录音或经任何信息存储检索系统对本书任何部分进行复制或传输。

本书中文简体版由中国机械工业出版社独家出版。

本书版权登记号: 图字: 01-2000-1414 号

图书在版编目 (CIP) 数据

UNIX 系统基础(版本 G.02)/(美)惠普公司(HEWLETT-PACKARD COMPANY)著;
陈谦等译. —北京:机械工业出版社,2000. 6

(HP-UX System Administration 认证辅导丛书)

ISBN 7-111-08134 -X

I. U… II. ①美…②陈… III. ①UNIX 操作系统②局部网络-基础知识 IV. TP393.1

中国版本图书馆 CIP 数据核字 (2000) 第 63173 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:王听讲 何月秋 封面设计:姚毅

责任印制:路琳

北京市密云县印刷厂印刷·新华书店北京发行所发行

2000 年 6 月第 1 版第 1 次印刷

787mm×1092mm 1/16 • 22.5 印张 • 537 千字

0 001—4000 册

总定价:120.00 元 (每册 40.00 元)

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话 (010) 68993821、68326677-2527

前 言

酝酿已久的 HP-UX 系列认证辅导丛书中文版终于同读者见面了。

HP-UX 作为企业级服务器、工作站上的主流 UNIX 操作系统，在全球范围的金融、电信、政府部门及其他行业的 IT 系统中早已扮演着举足轻重的角色。而随着 HP9000 惠普服务器、工作站在中国市场份额的日益扩大，全面地学习这一系统平台已成为我国众多 IT 工作者的迫切需要。

这套丛书译自惠普公司指定的英文版 HP-UX 培训教材。为了尽量保持原书风格，我们保留了原书的全部幻灯片。这样不仅使其线索更加清晰，而且大大方便了读者。事实上，在第一遍浏览时，你只需大体读懂每一页幻灯片，就等于抓住了内容的主干，而后再有选择的细看其下的注释说明，就会事半功倍。

此套丛书共分 3 册，它们循序渐进，自成体系。缺少 UNIX 使用经验的读者可以从《UNIX 系统基础》一书起步。在此基础上再学习《HP-UX 系统和网络管理 I》，因为它是学习绝大多数 HP-UX 课程的先行课。而《HP-UX 系统和网络管理 II》则侧重介绍基于 HP-UX 的网络原理、网络服务与配置。全面学完 3 册丛书后，你不但对 HP-UX 已经有了系统理解，而且对学习其他各种 UNIX 版本甚至非 UNIX 的操作系统平台，也定会有触类旁通、收益匪浅之感。

同时，此次出版的 3 册丛书也是针对惠普技术认证考试最具权威性的技术资料。认真学习这 3 册丛书，辅以一定的上机练习，你就很有希望获得新近推出、全球认可的惠普认证 HP-UX 技术专家 (HP CERTIFIED IT PROFESSIONAL ON HP-UX) 称号。你也可以在 www.hp.com.cn/support/education 找到惠普认证的相关信息和模拟考题。祝愿您能早日获得这一称号！

中国惠普公司培训事业部

2000 年 6 月

概 述

课程描述

该课程是惠普提供的第一门 UNIX 教程，其目的是让学习者（无论是系统管理员、编程人员还是一般用户）对 UNIX 有一个总体印象。该课程假设学习者对 UNIX [UNIX 是美国以及其他国家开放组织（Open Group）的注册商标] 或其他基于 UNIX 的系统均不了解。

学员能力目标

完成该课程的学习后，你将掌握以下内容：

第 1 章 UNIX 的介绍

- 描述 UNIX 操作系统的基本结构和功能。
- 描述 HP-UX。

第 2 章 登录及 UNIX 的基本使用

- 登录到 UNIX 系统。
- 退出 UNIX 系统。
- 在 HP-UX 参考手册中查阅命令。
- 通过在线手册查阅命令。
- 描述 shell 的命令行格式。
- 使用一些简单的 UNIX 命令识别系统使用者。
- 使用一些简单的 UNIX 命令与系统使用者通信。
- 使用一些简单的 UNIX 命令实现输出及其他用途。

第 3 章 CDE 的使用

- 描述前面板元素（Front Panel Elements）。
- 理解前面板的弹出菜单是如何工作的。
- 描述工作区切换台。
- 描述子面板控件。
- 明白如何使用帮助系统。

- 描述文件管理器。
- 知道如何使用文件管理器菜单。
- 用文件管理器定位文件。
- 删除文件。
- 用前面板、文件管理器和打印管理器打印文件。
- 显示打印缓存系统信息。
- 理解打印机管理器。
- 使用文本编辑器。
- 用应用管理器运行应用程序。
- 使用 Mailer 和 Mailer 选项，以及如何创建邮箱。
- 用日历管理器为日程和要做的事项安排时间。
- 描述如何浏览网络上其他的日历。
- 描述如何禁止或允许别人访问你的日历。

第 4 章 遍历文件系统

- 描述 UNIX 文件系统的结构。
- 描述文件和目录的区别。
- 成功地遍历文件系统。
- 建立和删除目录。
- 描述绝对和相对路径之间的区别。
- 使用相对路径名最大限度地减少打字量。

第 5 章 文件管理

- 使用常用的文件管理命令。
- 解释采用行式打印机缓存系统的目的。
- 鉴别和使用行式打印机缓存系统的命令与系统交互。
- 监控行式打印机缓存系统的状态。

第 6 章 文件访问权限

- 描述和改变文件的所有权和所在组这两个属性。
- 描述和改变文件的权限。
- 描述和建立新文件的默认权限。
- 描述如何改变用户和组的身份。

第 7 章 shell 的基础知识

- 描述 shell 的作用。
- 描述某用户登录时发生的事情。
- 描述用户环境的参数及其功能。
- 设置和修改 shell 参数。
- 理解并能修改特定的环境参数，如 PATH 和 TERM。

- 配置用户环境，使之能够配合某特定的应用程序。

第 8 章 shell 的高级特征

- 使用 shell 的各种替换功能。
- 设置和修改 shell 的参数。
- 把本地参数输出到环境中。
- 使参数能为子进程所用。
- 解释进程是如何产生的。

第 9 章 文件名生成

- 在命令行模式上使用文件名生成字符集生成文件名。
- 使用文件名生成字符集保存你的录入。
- 命名文件使文件名生成字符集更有效。

第 10 章 引 用

- 在命令行上使用 quoting（引用）机制而不去考虑命令中这些特殊字符本身的意思。也就是说这些字符本身被引用为别的用途。

第 11 章 输入和输出重定向

- 改变 UNIX 系统命令输出的目标文件。
- 改变 UNIX 系统命令生成的错误信息输出的目标文件。
- 改变 UNIX 系统命令输入的源文件。
- 定义过滤器。
- 会使用一些基本过滤器命令，像 sort、grep、和 wc。

第 12 章 管 道

- 准确描述管道的用法。
- 建立一个管道从一个指令中获得输出项，然后将此输出项作为其他指令的输入项。
- 会使用 tee、cut、tr、more 和 pr 等过滤器。

第 13 章 使用网络服务

- 能够描述 HP-UX 中各种不同的网络服务。
- 能够解释局域网的功能和作用（LAN）。
- 从局域网中找到本地系统和其他系统中的某个主机名。
- 能够使用 ARPA/Berkeley 服务实现远程登录、远程文件传输和远程执行指令。

第 14 章 vi 编辑器介绍

- 能够熟练地使用 vi 命令来编辑文本文件。

第 15 章 进程控制

- 会使用 ps 命令。
- 在后台启动一个进程。
- 使用 ps 命令监控正在运行的进程。
- 启动一个不受挂起信号影响的后台进程。
- 学会将后台进程切换到前台。
- 挂起一个进程。
- 发送信号终止正在运行的进程。

第 16 章 shell 编程介绍

- 编写基本的 shell 程序。
- 学会使用 shell 编程参数和环境变量。
- 学会使用 shell 编程参数和本地变量。
- 学会使用特殊的 shell 变量 * 和 #。
- 学会使用 shift 和 read 命令。

第 17 章 shell 编程——分支

- 描述条件性分支返回代码的作用。
- 用 test 命令分析一个命令的返回码。
- 在 shell 程序的分支中使用 if 和 case 结构。

第 18 章 shell 编程——循环

- 使用 while 结构，在某条件保持为真的情况下重复执行一段代码。
- 使用 until 结构，重复执行一段代码，直到某条件为真。
- 使用 for 结构，遍历由空白区分隔的一串参数。

第 19 章 文件脱机存储

- 用 tar 命令将文件存储到磁带上。
- 用 find 和 cpio 命令将文件存储到磁带上。
- 用 tar 或 cpio 命令恢复已被存储的文件。

附录

- 常用命令及其正确使用方法。

第 1 章 UNIX 的介绍

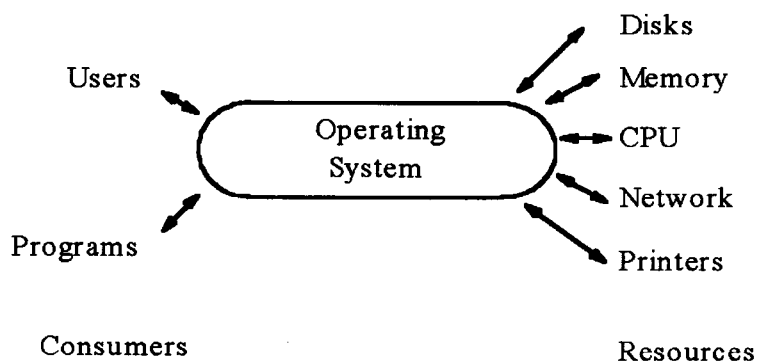
目标

完成此章学习后，你将掌握下列内容：

- 描述 UNIX 操作系统的基本结构和功能。
- 描述 HP-UX。

1.1 什么是操作系统

What Is an Operating System?



注释

操作系统是一种控制计算机（硬件）的特殊计算机程序（软件），在计算机和资源消费者之间起到联接的作用，常常用来在多消费者使用的情况下协调分配有限资源。这些资源包括 CPU、磁盘、内存和打印机等，消费者运行程序时需要访问这些资源。如当使用者或者程序提出请求要求在磁盘存储文件时，操作系统就会介入，分配好文件存储的磁盘空间，然后把文件信息从内存转到磁盘上。

当使用者提出执行程序的请求时，操作系统必须分配内存空间，以便载入和访问程序。

执行程序时，它会访问中央处理器（Central Processing Unit, CPU）。在一个分时系统中，常常有几个程序同时要求访问 CPU。

就像警察在一个复杂的十字路口指挥交通一样，操作系统决定程序如何轮用 CPU，以及何时占用 CPU。十字路口正如 CPU，接入十字路口的道路就像一个程序。在任何时间只有一条道路上的车辆能通过十字路口，警察指定哪条道路的车辆能通过十字路口，最终使所有道路的车辆都能顺利通过十字路口。

1.2 UNIX 操作系统的历史

History of the UNIX Operating System

Late 1960s	AT&T development of MULTICS
1969	AT&T Bell Labs UNIX system starts
Early 1970s	AT&T development of UNIX system
Mid 1970s	University of California at Berkeley (BSD) and other universities also research and develop UNIX system
Early 1980s	Commercial interest in UNIX system DARPA interest in BSD Hewlett-Packard introduces HP-UX
Late 1980s	Development of standards Open Software Foundation (OSF) founded
Early 1990s	POSIX, standardization of the interactive user interface

注释

1969年，UNIX操作系统起源于贝尔实验室。贝尔实验室的工程师 Ken Thompson 在 Rudd Canaday、Doug McIlroy、Joe Ossana 和 Dennis Ritchie 的协助下，编写了一个能实现普通应用共享时段的小系统，这个系统开始引起人们的注意。这些早期的开发人员许诺向实验室的管理人员提供各种优秀的文档编辑工具，因此得到一个更大的计算机，继续进行开发。

在 20 世纪 70 年代中期，许多大学得到了 UNIX 系统的使用许可证，UNIX 系统在学术领域受到热烈的欢迎，原因如下：

- 系统小 早期开发出的系统仅要求 512K 磁盘空间，其中系统占用 16K，用户程序占 8K，文件占 64K。

- 灵活 人们可以获取源代码,且该系统用高级语言编成,可移植性比较高。
- 价格低廉 各大学仅需付一个磁带的价钱就可以取得 UNIX 系统的使用许可证。

当然,早期 UNIX 系统提供的一些强大功能只能在更昂贵的硬件上实现。

这些优点抵消了系统当时的一些缺点:

- 没有技术支持 AT&T 在 MULTICS 上的投入已经相当多,没有兴趣研究 UNIX 操作系统。

- 有漏洞 既然没有技术支持,当然没有人保证消除系统的漏洞。

- 没有或者几乎没有文档资料,但人们可以从源代码获取信息。

当 UNIX 进入位于加利福尼亚大学伯克利分校时,伯克利分校的使用者开发了自己的操作系统版本。在美国国防部的支持下,他们加入许多新的特性。作为一个研究机构,伯克利分校对其被授权者采取与 AT & T 类似的技术支持政策 —— 没有技术支持!

AT&T 后来认识到这个操作系统的潜力,开始以商业方式发行 UNIX 使用许可证。为了加强他们的产品,AT&T 将其各部门完成的内部开发成果整合起来,同时开始将伯克利分校开发的增强特性融进他们的产品。

他们后来取得了成功,这些成功之处在于:

- 用户界面灵活,操作系统的工作环境包括了大量的软件工具。
- 系统以模块化设计,允许增加新的工具。
- 支持并开发多用户、多进程。
- 得到伯克利分校国防部高级项目研究机构(Department of Defense Advanced Research Projects Agency)的支持。

- 人们能够得到功能比以前强、价格比以前低廉的微机。
- 人们获得的 UNIX 系统与很大范围内的硬件平台兼容。
- 接口定义的标准化提高了应用程序的可移植性。

1.3 参考内容: UNIX 操作系统的历史

以下是 UNIX 历史更详细的资料:

- 1956 AT&T Consent Decree 颁布,这个针对 AT&T 的反托拉斯法案禁止 AT&T 进入一些未受管制的领域。

- 1965 贝尔实验室与麻省理工学院开始在最终实现多用户的环境中共同研究开发多线性信息与计算系统 (MULTIplexed Information and Computing System, MULTICS)。

- 1969 在 UNIX 系统的诞生地 —— 贝尔实验室,工程师 Ken Thompson 在对文件系统的研究过程中,在一台废置的 PDP-7 小型计算机上编成一个模拟人体空中运动的程序,叫空中旅行 (Space Travel)。他还编了一个文件系统、一个汇编器、文本编辑器和一个简单的 shell。为什么要用 PDP-7 呢?原因是它能提供良好的图形,且比起支持可交互、分时的 DEC-10 要便宜,但 Ken Thompson 需要的是能提供方便的可交互性服务的计算机(以前他在 GE645 上做研究开发,GE645 用批处理的方式工作,访问资源的代价昂贵)。写好的程序

首先被 PDP-7 交叉编译好后, 通过纸带传进去。由于 Consent Decree 的原因, 贝尔实验室只可以研究 UNIX 系统, 但不能销售、宣传或者支持任何基于 UNIX 的产品。他们仅可向大学发行软件, 供科学研究用。

- 1970 基于汇编语言的 UNIX 系统被移植到 PDP-11/20 (16 位小型计算机) 上开发文字处理功能。

- 1971 UNIX 第 1 版诞生, 贝尔实验室专利局成为第一个用户。由于使用者无须采用中央计算服务, 显示了很大的优越性。Ken Thompson 在 Martin Richards 的 BCPL 语言基础上开发了解释语言 B, 后来接着开发了 NB 语言 (new B)。

- 1972 UNIX 第 2 版诞生, 新特征有: 管道、支持编程语言、开始尝试用 NB 来编写内核 (NB 是 C 语言的前身)。人们使用的 UNIX 系统个数达到 10 个。Dennis Ritchie 开发了 C 语言。

- 1973 UNIX 第 4 版诞生, 内核和外壳用 C 语言重写而成。贝尔实验室的 UNIX 系统小组 (UNIX System Group) 成立, 对 UNIX 研究开发做内部的技术支持。人们使用的 UNIX 系统个数达到 25 个。第一次向大学发行非正式版。

- 1974 UNIX 第 5 版诞生, 向大学发行正式版作科研使用。AT&T 不提供技术支持, 没有试用期限, 没有质量保证, 不许诺修补漏洞, 而且要求预先付款购买许可证。

- 1975 UNIX 第 6 版诞生, 开始向政府和商业用户发行使用许可证。Thompson 在休假年加入加利福尼亚大学伯克利分校。伯克利分校开始研究开发 UNIX。

- 1977 人们使用的 UNIX 系统个数达到 500 个, 其中大部分分布在 125 所大学。人们开始在 PDP-11 上研究开发成功第一个 BSD 操作系统。UNIX 系统第一次被移植到非 DEC 机上。

- 1978 UNIX 第 7 版诞生, 设计时主要着眼于提高可移植性, 能支持交换区、K&R C 编译器、Bourne shell 以及更大的文件。UNIX 系统被移植到 VAX11/780 (支持 32 位地址空间和 4G 的虚拟地址空间), 移植后诞生了 UNIX/32V。

- 1979 BSD3.0 诞生, 这是 UNIX/32V 的增强版, 加入了对虚拟内存和按需分页的支持。它的主要设计目标是能运行所需内存比物理内存大的进程。

- 1980 BSD4.0 诞生, 加入了作业管理、虚拟内存、分页、第三方 (非 DEC) 外围设备的驱动程序, 实现了基于屏幕的应用程序, 如 vi, 与终端类型无关。这个操作系统引起 DARPA 的兴趣, DARPA 当时正在为其 CAD/CAM、人工智能、视觉应用的网络化研发环境寻找一个开放的操作系统标准。伯克利分校对虚拟内存的研究超过了 AT&T。

- 1981 /usr/group 成立, 这是第一个发起制定 UNIX 环境标准的组织。

- 1982 System III 诞生, 汇合了 AT&T 开发的几个 UNIX 变种的特征, 同时加入了 BSD 的一些特性, 如 curses、作业管理、termcap 和 vi。HP-UX 出现。

- 1983 System V 第 1 版诞生, AT&T 宣布正式支持该系统, 并降低了价格。AT&T 对支持这个 UNIX 系统的微处理器生产厂家给予了授权。

基于 DARPA 研究开发成果的 BSD4.2 发布, 加入了对 IPC、虚拟内存、高速文件系统、网络协议 (TCP/IP) 的支持。16 位和 32 位的微机出现。人们使用的支持 BSD-IPC、网络协议和高速文件系统的 UNIX 系统达到 100 000 个。

- 1984 Consent decree 被废除, 贝尔实验室从 AT&T 分离出来, AT&T 被允许参与

计算机市场的竞争。

System V 第2版诞生, 支持分页、内存共享。

/usr/group 制定的标准向 POSIX 提交。

- 1985 System V Interface Definition(SVID)诞生, 定义了系统调用的接口。System V Verification Suite(SVVS)诞生, 只有通过这个测试套装才能符合 SVID。

- 1986 BSD4.3 诞生, 与前版不同之处主要是修正了一些漏洞, 作业管理功能增强, 信号机制更加可靠。

- 1987 System V 第3版诞生, 支持 STREAMS、IPC、作业管理。

X/Open Portability Guide(XPG)标准发布, 对内核接口和许多应用程序进行定义, 这个标准提高了应用程序在不同 UNIX 系统之间的可移植性。300 000 套 UNIX 系统软件出口到国外。人们使用的 UNIX 系统达到 750 000 个。

- 1988 SVID 第2版 —— 增加了文件锁定的标准。

Open Software Foundation 成立, 这是个独立的公司, 成立的目的是建立和提供一个以工业标准和当前可以获得的最先进技术为基础的计算环境。

- 1989 System V 第4版诞生, 遵从 POSIX.1。

XPG/3 发布, 支持 POSIX.1 和通用应用程序环境标准(Common Application Environment), 在提高可移植性方面, 增加了计算环境几方面的标准, 而不仅仅局限于定义操作系统接口。

- 1990 SVID 第3版发布, 支持 POSIX.1、FIPS 151-1 和 C 标准。

- 1991 HP-UX 8.0 面世, 取得系统 5 第3版的认证, 遵从 SVID2, 将实际上已经成为业界标准的 BSD4.2 和 BSD4.3 的扩展特性包括进去, 建立了支持 POSIX-、FIPS-、XPG2-和 XPG3-的接口。

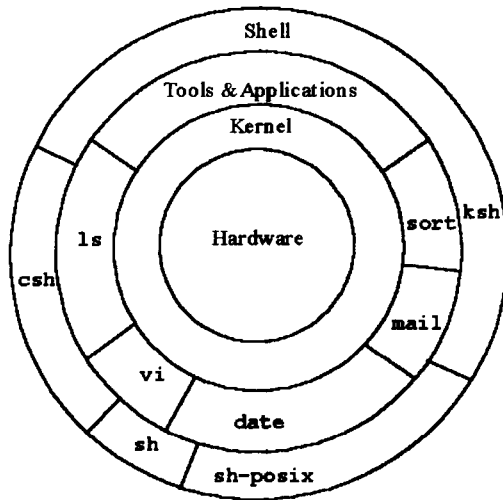
- 1992 HP-UX 9.0 面世, 取得系统 5 第3版的认证, 遵从 SVID2, 支持 X/Open Portability Guide 第3版、POSIX 1003.1、POSIX 1003.2、X11R5、FIPS-2、FIPS-3、POSIX.1、OSF/Motif1.2 和其他的标准。

- 1995 HP-UX 10.0 面世, 在内核遵从 SVID3, 支持 X/Open Portability Guide 第4版、POSIX .4 Realtime Phase 1, 还有保证程序从 9.0 转至 10.0 的可移植性部件。与上一版的主要区别是文件系统的结构改为支持 AT&T SVR4 和 OSF/1。

- 1997 HP-UX 11.0 面世, 遵从 SVID 4 和 POSIX.2。在对 64 位应用程序的支持方面, 符合 IA64 标准, 实现内核线程化。

1.4 UNIX 的特征

Features of UNIX



注释

UNIX 系统通过时段共享, 控制计算机的活动和资源, 同时提供了一个灵活的接口。UNIX 最初的设计目的是为了同时运行多线程和支持多用户, 从而使项目组成员之间共享数据变得更方便。UNIX 系统在所有的层次上都采用了模块化结构。安装 UNIX 系统时, 可以按需安装, 不必安装多余的部件。例如, UNIX 系统提供了大量的编程工具, 但你如果不做编程, 只需安装最少的部件。用户接口同样鲜明地体现了这种模块化的思路。可以用管道将彼此互不相关的命令联接起来实现相当复杂的管理功能。

操作系统

内核就是操作系统。它负责对可用资源进行管理和对硬件进行访问控制。内核包含它要交互的每一个硬件部件的模块, 通过这些模块程序可以访问 CPU、内存、磁盘、终端和网络等等。当新的硬件安装在系统上时, 新的模块就被嵌入到内核中。

操作环境

1. 工具和应用程序

UNIX 的模块设计思路在这一层体现得最明显。UNIX 系统命令的原理是：每个命令能很好地完成一个任务，多个命令结合起来形成一个工具箱。当你想完成一个作业时，可以抽出合适的工具来实现它，还可以通过正确的工具组合完成一个复杂的任务。

从一开始，UNIX 系统“工具箱”包含的命令就远远多于与系统交互所必需的基本命令。UNIX 系统还为以下功能提供了命令：

- 电子邮件 (mai、mailx)
- 文件编辑 (ed、ex、vi)
- 文本处理 (sort、grep、wc、awk、sed)
- 文本格式化 (nroff)
- 程序开发 (cc、make、lint、lex)
- 程序管理 (SCCS, RCS)
- 同类型系统之间的通信 (uucp)
- 进程与用户帐号管理 (ps、du、acctcom)

由于 UNIX 系统的用户环境被设计成为一个可交互、可修改的模块化环境，因此可以很容易地开发和增加新的应用到使用者的工具箱中，不需要的工具可以删除，整个系统的运作不会受到影响。

例如，一个应用程序开发人员和一位专业作家都在用 UNIX 系统工作，尽管其专业领域非常不同，但他们都使用许多相同的命令。他们也要用一些与各自专业相关的应用程序。应用程序开发人员会使用有关程序开发和管理的工具，而作家会采用文本格式化、文本处理和文档管理的软件。有趣的是，应用程序开发人员使用的程序修改工具，作家也可以用来修改文章。因此，尽管他们的系统看上去非常相似，事实上双方都根据他们的使用需要选择了相应的模块。

UNIX 系统之所以受欢迎，是因为其具有以下特点：

- UNIX 系统的完整性和灵活性使它能与许多应用环境相适应。
- 操作环境中自带大量的工具，提高了使用者的工作能力。
- 可移植性高，在许多硬件平台上都可以运行。

2. shell

shell 是一个命令解释器。在命令提示符输入命令，发出后会被执行。使用者通过 shell 与计算机通信。shell 接收用户在键盘上键入的内容，并把这些内容翻译成内核可以理解的形式，然后系统执行这个命令。

应该注意到 shell 是与内核分离的。如果你不喜欢系统提供的 shell 界面，可以很轻松地用其他的 shell 代替它。现在有很多 shell 可供选择。有的 shell 是命令行方式，有的是菜单方式。一般来说，UNIX 系统自带的 shell 既提供命令解释器，又提供编程用的接口。

目前 UNIX 系统有四种 shell 可供选择，其分别是：

- Bourne shell (/usr/old/bin/sh) AT&T UNIX 系统最初提供的 shell，由贝尔实验室

的 Stephen Bourne 开发而成。它提供了 UNIX 的命令解释器和编程用的接口，该接口可用来开发 shell 程序，常常被称为脚本（script）。这些可编程的交互式接口可以实现变量定义和替换、变量和文件检测、分支和循环等功能。

- C shell (usr/bin/csh) 基于 BSD 的 UNIX 系统提供的 shell，由加利福尼亚大学伯克利分校的 Bill Joy 开发而成。人们将这个 shell 称为 California shell，缩短拼写后就成了 C shell。由于它具有一些交互性的特征，如可以从命令栈恢复和编辑曾经键入的命令以及别名机制（允许使用个性化的名字取代系统当前的命令），人们认为它比 Bourne shell 更先进了一步。

- Korn shell (/usr/bin/ksh) 这是贝尔实验室较新的开发成果，由 David Korn 开发而成。它支持 Bourne shell 简单的编程接口，因此可以认为是 Bourne shell 的增强版，但它同时具有 C shell 方便、可交互的特性。这个 shell 的代码已被优化以提供更快、更高效率的 shell。

- POSIX shell (/usr/bin/sh) 遵从 POSIX，包括编程语言和命令解释器，位于 /usr/bin/sh。这个 shell 与 Korn shell 在很多方面很相似。它提供历史命令机制，支持作业管理，还具有其他各种有用的特点。各种 shell 的特征比较见表 1-1。

表 1-1 各种 shell 的特征比较

特 征	描 述	Bourne	Korn	C	POSIX
历史命令	以前使用过的命令可以存在缓存中做修改或重新调用	否	是	是	是
行编辑	可以用文本编辑器修改当前或以往的命令	否	是	否	是
文件名补齐	在命令行中自动补齐文件名	否	是	是	是
别名命令	可以用来重命名命令、自动包括命令选项和精简长命令行	否	是	是	是
限制版 shell	着眼于安全方面的特性，其环境的功能受到限制	是	是	否	是
作业控制	跟踪和访问在后台执行进程的工具	否	是	是	是

1.5 UNIX 的其他特征

More Features of UNIX

- Hierarchical file system
- Multi-tasking
- Multi-user

注释

层次性的文件系统

存储在磁盘上的信息放在某种容器上，这种容器叫做文件。每个文件有一个名字，用户通过指定名字访问文件。文件通常包含数据、文本、程序等等。一个 UNIX 系统一般有数以百计的文件，用户可以用另外一种容器——目录，将他们的文件组织成为一个个逻辑组。在 UNIX 系统中，目录可以用来存储文件或其他目录。

文件系统的结构非常灵活，因此当用户组织信息的需求改变时，可以用简单的 UNIX 系统命令把文件和目录移动、重命名或者组成新的或另外的目录。所以文件系统就像一个电子文件柜一样。文件系统允许用户分离和组织他们的信息，把信息置于对其环境和应用最合适的目录中。

多任务系统

在 UNIX 系统中可以同时执行多个任务。就单独一个终端而言，用户可以执行几个任务，但这几个任务看起来在同时执行。这就意味着，用户在编辑一个文本文件时，可能此时另一个文件正在被格式化，同时，另外一个文件正在被打印。

事实上，CPU 在某个时候只能执行一个任务，但 UNIX 操作系统能够将 CPU 的时间分片，这些片分配给被安排同时执行的多个进程。因此，从使用者看来，所有的程序都在同时运行。

多用户系统

UNIX 对多用户的支持使得多个用户能在同一时间登录和使用系统，多个终端和键盘可以同时与同一台计算机建立联接。这是 UNIX 支持多任务的功能自然延伸的结果。如果系统能同时运行多个应用程序，那么这些应用程序有一部分应该能够被其他用户的进程所用。另外，一个用户能够在不同终端多次登录同一个系统。无论从开发小组还是某个使用者的角度来看，这种支持多用户的功能都将显示出一个很大的优越性：一个工作组的所有成员能同时访问同一些数据。